Solar Phys (2008) 248: 359–377
DOI 10.1007/s11207-007-9064-9

Comparison of Five Numerical Codes for Automated Tracing of Coronal Loops

Markus J. Aschwanden · Jong Kwan Lee · G. Allen Gary · Michael Smith · Bernd Inhester

Received: 21 May 2007 / Accepted: 27 September 2007 / Published online: 13 November 2007

Abstract The three-dimensional (3D) modeling of coronal loops and filaments requires algorithms that automatically trace curvilinear features in solar EUV or soft X-ray images. We compare five existing algorithms that have been developed and customized to trace curvilinear features in solar images: *i*) the oriented-connectivity method (OCM), which is an extension of the Strous pixel-labeling algorithm (developed by Lee, Newman, and Gary); *ii*) the dynamic aperture-based loop-segmentation method (developed by Lee, Newman, and Gary); *iii*) unbiased detection of curvilinear structures (developed by Steger, Raghupathy, and Smith); *iv*) the oriented-direction method (developed by Aschwanden); and *v*) ridge detection by automated scaling (developed by Inhester). We test the five existing numerical codes with a TRACE image that shows a bipolar active region and contains over 100 discernable loops. We evaluate the performance of the five codes by comparing the cumulative distribution of loop lengths, the median and maximum loop length, the completeness or detection efficiency, the accuracy, and flux sensitivity. These algorithms are useful for the

M.J. Aschwanden (✉)
Solar and Astrophysics Laboratory, Lockheed Martin Advanced Technology Center, Department ADBS, Building 252, 3251 Hanover Street, Palo Alto, CA 94304, USA
e-mail: aschwanden@lmsal.com

J.K. Lee
Department of Computer Science, University of Alabama in Huntsville, Huntsville, AL 35899, USA
e-mail: jlee@cs.uah.edu

G.A. Gary
National Space Science and Technology Center, University of Alabama in Huntsville, Huntsville, AL 35899, USA
e-mail: Allen.Gary@nasa.gov

M. Smith
Mullard Space Science Laboratory, Holmbury St. Mary, Dorking, Surrey RH5 6NT, UK
e-mail: ms2@mssl.uc.ac.uk

B. Inhester
Max-Planck Institute for Solar System Research, 37191 Katlenburg-Lindau, Germany
e-mail: binhest@mps.mpg.de

reconstruction of the 3D geometry of coronal loops from stereoscopic observations with the STEREO spacecraft, or for quantitative comparisons of observed EUV loop geometries with (nonlinear force-free) magnetic field extrapolation models.

Keywords Corona: structures · Sun: EUV · Methods: pattern recognition

1. Introduction

Objective data analysis of solar images requires automated numerical codes that supplement arbitrary visual/manual pattern recognition by rigorous mathematical rules. Generally, solar images display area-like features (active region plages, sunspots, flare ribbons, partial-halo coronal mass ejections, *etc*.) as well as curvilinear features (coronal loops, filaments, prominence fine structure, *etc*.). In this paper, we focus on automated detection of curvilinear features that appear in solar images recorded in extreme ultraviolet (EUV), soft X rays, white light, or Hα (Aschwanden, 2004).

A recent review on two-dimensional (2D) feature recognition and three-dimensional (3D) reconstruction in solar EUV images is given in Aschwanden (2005). Solar EUV imaging started with the Extreme-ultraviolet Imaging Telescope (EIT) and Coronal Diagnostic Spectrometer (CDS) on the *Solar and Heliospheric Observatory* (SOHO), the *Transition and Coronal Explorer* (TRACE), the Extreme Ultraviolet Imager (EUVI) on the STEREO spacecraft, and the Extreme ultraviolet Imaging Spectrometer (EIS), and now includes also X rays with the X-ray Telescope (XRT) on the *Hinode* spacecraft and the future Atmospheric Imaging Assembly (AIA) on the *Solar Dynamics Observatory* (SDO).

The basic premise of an automated tracing algorithm is to achieve an objective method to identify and measure the geometry of curvilinear features. For coronal loops for instance, such a method can identify all detectable loops in an image and measure their spatial coordinates $[x(s), y(s)]$, their lengths L, and their curvature radii $R(s)$, as a function of a length coordinate s. Ideally, we want to trace loops from one to the other footpoint, but automated tracing may often be restricted to partial segments, missing parts with low densities (near loop tops or the cooler footpoints), or be misled by crossing loops in the foreground and background. Nevertheless, such a capability allows one to also count detectable loop structures in an image in an automated way, to track the same loop structure in a time sequence of images, and to detect time-dependent spatial variations, such as eigen-motions, velocities, oscillatory motions, and wave motions along loops. Such an algorithm can then be employed for blind searches of structures with selected velocities, oscillation periods, or amplitudes. For large data sets in particular, such as the upcoming SDO mission, automated detection methods will be inevitable to find structures of interest at all for a particular science study. Automated loop-tracing algorithms will be particularly useful for 3D reconstruction of loop geometries using STEREO data, as well as for constraining theoretical magnetic field extrapolation models (*e.g.*, see Wiegelmann and Inhester, 2006).

2. Automated Loop-Tracing Codes

There exist a large number of numerical, automated, pattern recognition codes, specialized either for area-like features or on curvilinear features. However, there exists no general pattern recognition code that has a superior performance in all kinds of data, so the best code for a particular data set needs to be tailored and customized to the particular morphological

properties of the given data set. Here we are mostly interested in the automated detection of soft-edge curvilinear features in solar images, and thus we largely ignore numerical codes that work on nonsolar images, such as used for analysis of medical, geological, or geographical images. To our knowledge, there exist so far five codes that are specialized for solar images, all of which were developed in recent years by four independent groups. We briefly describe the numerical algorithms of these five codes in this section and provide a comparison of their performance in Section 3.

2.1. The Oriented-Connectivity Method

The oriented connectivity-based method (OCM) (Lee, Newman, and Gary, 2004, 2006b) is the first automated coronal-loop segmentation method that performs its loop segmentation via a constructive edge linkage method. Since the coronal loops are the vestiges of the solar magnetic field, OCM's edge linkage is based on model-guided processing, which exploits external estimates of the magnetic field's local orientation derived from a simple dipolar magnetic field model. OCM includes three processing steps: *i*) preprocessing, *ii*) model-guided linkage, and *iii*) postprocessing.

The goal of the preprocessing step is to remove pixels that are very unlikely to be coronal loop pixels and to precompute the estimates of the magnetic field's local orientation. A median filtering and unsharp masking (*i.e.*, contrast enhancement) are applied to remove the image noise and to sharpen coronal loop structures. Strous' loop pixel labeling algorithm (Strous, 2000) is also used to determine the possible loop pixels followed by median-intensity thresholdings to reduce the number of falsely labeled loop pixels. In addition to these image "cleaning" steps, the magnetic field's local orientation is restricted by a dipolar magnetic field model by using a magnetogram, *e.g.*, from the Michelson Doppler Imager (MDI) onboard the SOHO spacecraft. In particular, the local orientation estimation exploits the multipolar field model by considering a set of estimates of the 3D-to-2D-projected magnetic fields' orientation.

Using the potential loop pixels and the magnetic fields' local orientation determined in the preprocessing step, the OCM performs coronal-loop segmentation via a constructive edge linkage guided by the magnetic field's orientation (*i.e.*, magnetic field local orientation estimates are used to progressively link loop pixels with consistent orientation). Starting from a loop pixel determined in a preprocessing step, the OCM segments loop pixels by forming a clustering of all the other pixels that define the same loop structure. This forming of a clustering of loop pixels is a stepwise process, which at each step adds one loop pixel to the current loop. The selection of the "best" loop pixel exploits a weighting scheme (*i.e.*, distance-, intensity-, angular-, and tangent-based weighting) among all the candidate pixels that are determined using the local orientation estimates.

In the postprocessing step, a (coronal loop) spline fitting and a linkage step followed by another spline fitting are applied to join the disconnected subsegments of the loops and remove the aliasing. The first B-spline fitting is designed to produce smooth loop structures and a simple linkage step and the second B-spline fitting is applied to merge disconnected loop segments smoothly.

2.2. The Dynamic Aperture-based Loop Segmentation Method

Carcedo *et al.* (2003) reported that the coronal loops have a cross-sectional intensity profile following a Gaussian-like distribution. The dynamic aperture-based method (DAM) (Lee,

Newman, and Gary, 2006b) exploits the Gaussian-like shape of loop cross-sectional intensity profiles and constructively segments credible loop structures. In particular, DAM segments coronal loops via a search through the image for regions whose intensity profiles are well-fitted by a ruled Gaussian surface (RGS). Since nearby RGSs that are on the same loop appear to have similar Gaussian shape parameters, DAM forms loops through a linkage process that clusters adjacent fitted RGSs if their shape parameters are similar. In addition, the clustering joins fitted RGSs only if they have similar orientation, which is determined by applying principal component analysis (*e.g.*, the arctangent of the maximum eigenvector components is used to estimate the loop angular direction) on the RGSs. DAM also includes preprocessing steps to remove nonloop structures and postprocessing steps to remove the aliasing and to join the disconnected loop segments. The preprocessing step, in particular, is designed to remove the image noise, to enhance the contrast between the loops and the background, and to remove nonloop structures by median filtering and high-boosting followed by a global mean-intensity thresholding. (The postprocessing steps used in DAM are the same as the postprocessing steps used in OCM.)

2.3. Unbiased Detection of Curvilinear Structures Method (UDM)

This code for unbiased detection of curvilinear structures, developed by Steger (1996), detects a curvilinear feature from the local edge contrast, while also taking the geometry of its surroundings into account. Essentially, the centroid position of a curvilinear structure is determined from the second derivative in the direction perpendicular to the curvilinear structure, which yields a stable position with subpixel accuracy that is independent of the width and asymmetric profile of the structure (and thus "unbiased"). Such tracings of curvilinear structures have been successfully applied to photogrammetric and remote-sensing images from satellites and airplanes (tracing roads, railways, or rivers), as well as to medical imaging (tracing blood vessels from X-ray angiograms or bones in the skull from CAT-scan or magnetic resonance images) (Steger, 1996). The code from Steger was further explored in the work of Raghupathy (2004), who optimizes the connection (linkage) or loop segments at junctions, crossovers, or other interruptions, using the eigenvalues and eigenvectors of the Hessian matrix of second-order derivatives, as well as a generalized Radon transform (essentially the integral of the image along a traced curve). Thus, Raghupathy's approach breaks the loop-tracing problem down into *i*) a local search of loop segments using the gradients among neighbored pixels and *ii*) a global search of identifying connections between disconnected curve segments using the generalized Radon transform. The algorithm of Steger (without Raghupathy's modification) has been rewritten in the computer language C by Smith (2005) for the SDO team at the Mullard Space Science Laboratory (MSSL) and applied to TRACE images.

2.4. Oriented-Directivity Loop Tracing Method

The oriented-directivity loop tracing method (ODM) is currently being developed by Aschwanden, with testing still in progress. The ODM code uses directional information for guiding the tracing of coronal loops, similar to the OCM code, but makes use only of the local directivity, whereas the OCM code estimates the global connectivity between the endpoints of the loops (by using *a priori* information derived from magnetic field models). The code consists essentially of four parts; *i*) preprocessing of the image to render the highest contrast for fine loops, *ii*) finding starting points of loops, *iii*) bidirectional tracing of the loop segments to the two opposite endpoints, and *iv*) reducing data noise by smoothing of loop

curvatures. In the preprocessing step, several high-pass filters are applied over the range of loop widths of interest (say, approximately three to seven TRACE pixels here). In the second step, an arbitrary starting point of a candidate loop is identified by searching for local (intensity) flux maxima in the high-pass-filtered image in each macropixel (say, with a size of ten pixels) of the full image. In a third step, a loop is traced by moving a search box (*e.g.*, of size ten $\times$ ten pixels) along the expected local direction in small steps (say, five pixels). The local direction is determined by performing a linear regression fit to the flux maxima positions inside the search box, where local maxima with weak flux below some threshold and with positions outside the expected direction (within some tolerance, say, $\pm$ three pixels) are ignored. The direction of a traced loop is then updated at each subsequent position and used for extrapolating the next loop position. If no suitable direction is found or if the flux is below the threshold, the loop segment is terminated. The loop tracing is conducted in both directions from the initial starting point to obtain a full loop segment. In the last step, the curvature of each loop segment is smoothed with a median filter to reduce the inherent data noise. The ODM code was applied to TRACE images and compared with visual/manual loop tracings.

2.5. Ridge Detection by Automated Scaling

The ridge detection by automated scaling (RAS) method can be considered as an extension of the OCM by Lee, Newman, and Gary (2006a) and was developed by Inhester. The improvements consist of replacing Strous' ridge point detection scheme by a modified multiscale approach of Lindeberg (1998), which automatically adjusts to varying loop thickness and also returns an estimate of the reliability of the ridge point location and orientation (Inhester, Feng, and Wiegelmann, 2007). The connection of loop segments is accomplished by geometrical principles that include the orientation of the loop at the ridge points with the co-circularity constraint proposed by Parent and Zucker (1989). The code consists of three modules: *i*) search for ridgel location, *ii*) ridgel connection to chains or more complete loops, and *iii*) curve fits to the ridgel chains to obtain smooth, spline-represented loop curves. In the first step, the determination of the characteristic points at ridge centers is computed from the Taylor coefficients of the local regularized differentiation, where the optimum regularization parameter is determined from a maximum of the quality function $q = d^{\gamma}(|h_1| - |h_2|)$. This function q depends on the window size d and the eigenvalues h_i of the second-derivative matrix, which for ridges has to satisfy $h_2 < h_1$. The Taylor coefficients interpolated at the center of the ridge yield the position, orientation, and quality of the "ridgel." In the second step, ridgels are combined into chains, where all possible ridgel – ridgel connections are weighted according to their "binding energy," specified by the mutual distance and orientation, the latter being quantified by the co-circularity measure of Parent and Zucker (1989). In the third step, ridgel chains are smoothly connected by polynomial fits, which balance curvature, distance, and orientations. The mathematical framework of this RAS code and an application to first EUVI/SECCHI images has been described in Inhester, Feng, and Wiegelmann (2007).

3. Test Comparisons of Loop-Tracing Codes

In this section we compare the five different codes described here by applying them to the same test image, which was chosen to be the same as used in some earlier code demonstrations (Lee, Newman, and Gary, 2004, 2006a, 2006b; Smith, 2005).

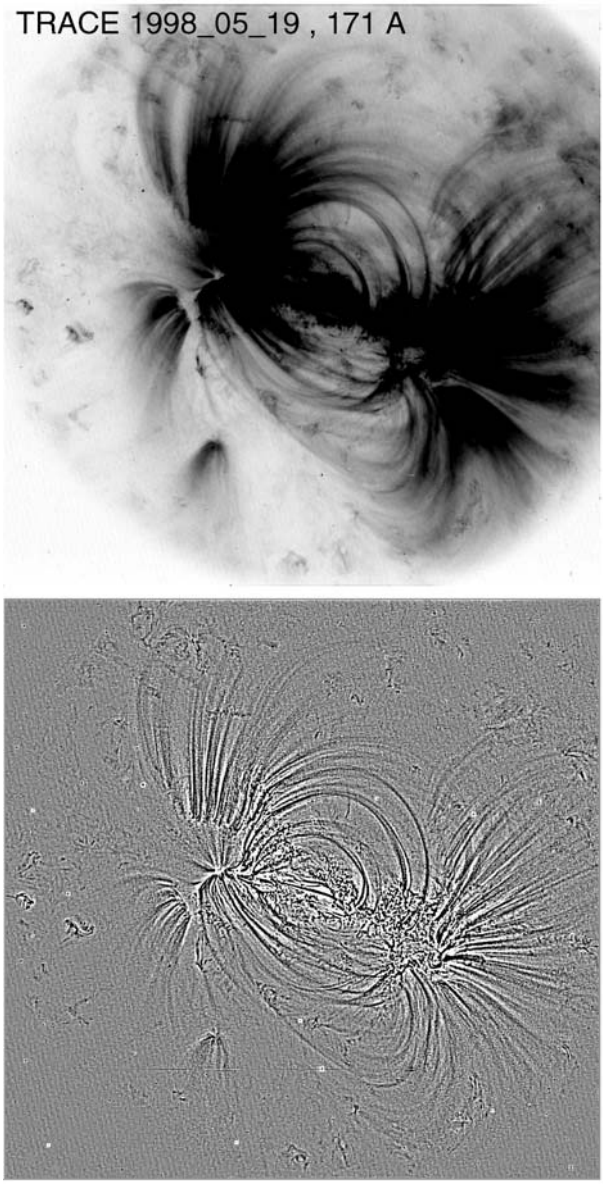

Figure 1 Top: Original 171 Å TRACE image of 19 May 1998, 22:21 UT, with a size of 1024×1024 pixels. Black means high fluxes. The circular white boundary is caused by the vignetting of the telescope (not to be confused with the solar limb). Bottom: High-pass-filtered TRACE image, where a 7×7-boxcar smoothed image was subtracted from the original image. The background also reveals some residual nonsolar spikes, JPEG compression artifacts, and diagonal quasiperiodic ripples caused by electromagnetic interference (EMI) in the CCD readout.

3.1. Test Image and High-Pass Filtering

We show the test image in Figure 1, which is an EUV image observed with the TRACE telescope on 19 May 1998, 22:21:43 UT, with an exposure time of 23.172 seconds in the wavelength of 171 Å. The image consists of 1024×1024 pixels, with a pixel size of $0.5''$. The pointing of the TRACE telescope was near disk center (*i.e.*, the Sun center is at XCEN $= 422.027$ and YCEN $= 401.147$ pixels), which is slightly southeast of the image center [512, 512]. For count statistics and contrast we report the average flux value in the image, which is 146 ± 62 DN (data numbers), with a minimum value of 56 DN and a max-

imum of 2606 DN. The image has been processed for flat-fielding and spike removal (with the standard TRACE_PREP procedure).

The original image is shown on a logarithmic gray scale in the top frame of Figure 1, using an inverted gray scale (where black means high fluxes). A high-pass-filtered version is shown in the bottom frame of Figure 1, where a smoothed image (with a seven × seven boxcar) was subtracted from the original image to enhance the fine structure of coronal loops. The image reveals an active region with a fairly dipolar magnetic field, where over 100 individual coronal loops (or segments) are visible, mostly bright in the lowest vertical density scale height (of $\lambda \approx 50\,000$ km at a plasma temperature of $T = 1.0$ MK) above the solar surface.

3.2. Manual Tracing of Loops

The 171 Å and 195 Å TRACE images are most suitable for coronal loop tracing. To enhance the finest loop strands we apply first a high-pass filter by subtracting a smoothed image $I_S(x, y)$ from the original image $I_O(x, y)$, a method that is also called "unsharp-masking." The high-pass-filtered image $I_F(x, y)$ is defined as

$$I_F(x, y) = I_O(x, y) - I_S(x, y), \tag{1}$$

where a boxcar smoothing is applied with typically $N_{sm} = 5, 7, 9$, or 11 pixels to $I_S(x, y)$ (see the example in Figure 1, bottom, or the enlargement in Figure 2, bottom). The filtered image $I_F(x, y)$ is then enlarged by a factor of two or three on the computer screen to enable visual/manual tracing with subpixel accuracy (*i.e.*, the position of the spline points). Using an interactive cursor function, one then selects a number of spline points $[x_i, y_i]$, $i = 1, \ldots, n$, on the enlarged screen display, with typically $n = 5$ for short loops or $n = 10$ for long loops. A higher resolution of the loop coordinate points is then obtained by interpolating the spline coordinate points $[x_i, y_i]$, $i = 1, \ldots, n$, with a two-dimensional spline function (called SPLINE_P in IDL) with typically $N = 8n$ times higher resolution, yielding $N \approx 40$ coordinate points for short loops, or $N \approx 80$ coordinate points for longer loops. The two-dimensional cubic spline fit usually follows the curvature of the traced loops much more smoothly than manual clicking. Since short loops are nearly semicircular, three spline points would be the absolute minimum to interpolate a circle with constant curvature radius, and five to ten spline points allow us to follow steady changes of the curvature radius with sufficient accuracy without having redundant spline points. In the TRACE image shown in Figure 2 (bottom), we identified some 210 individual loops (Figure 2, top).

This method has been used to trace coronal loops in EIT 171, 195, and 284 Å data (Figure 4 in Aschwanden *et al.*, 1999a; Figure 1 in Aschwanden *et al.*, 2000a) and in TRACE 171, 195, and 284 Å TRACE data (Figure 1 in Aschwanden *et al.*, 1999b, 2002; Figures 3a – 3d in Aschwanden, Nightingale, and Alexander, 2000b; Figure 8 in Aschwanden and Alexander, 2001; Figures 4 – 8 in Yan *et al.*, 2001; and in Aschwanden and Nightingale, 2005).

3.3. Automated Tracing of Loops

In Figures 3 – 5, we show the results of automated loop tracings with the five codes described in Sections 2.1 – 2.5, with the same field of view as in Figure 2, which covers a subimage with pixel ranges of $x = 200 - 1000$ and $y = 150 - 850$ from the original 1024×1024 image shown in Figure 1. The same TRACE image (Figure 1, top) was given to the four originators of the five codes, with the only instruction being to provide the loop coordinates (x_i^L, y_i^L)

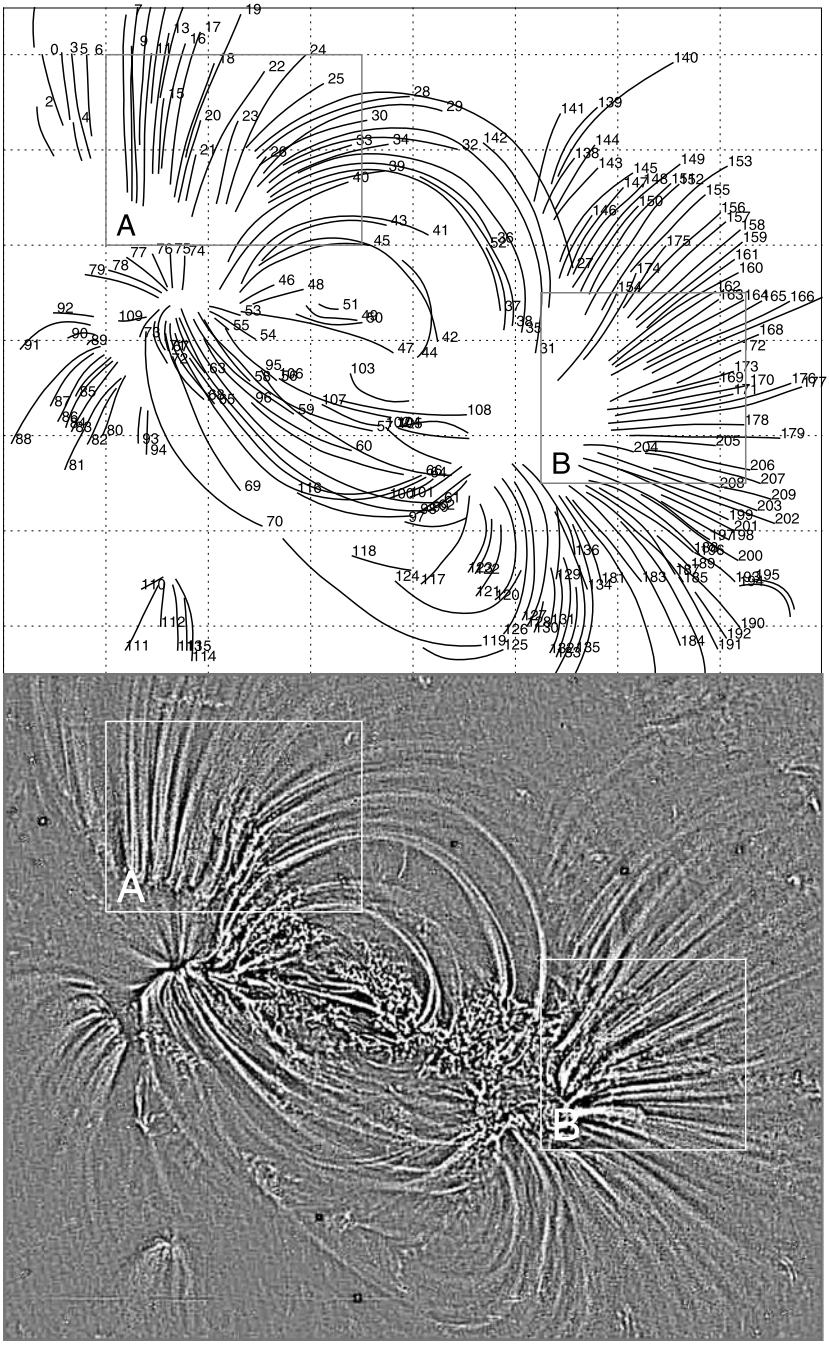

Figure 2 Top: Enlarged central part of the TRACE image (x-pixel range of 200 – 1000, y-pixel range of 150 – 850) showing the 210 manually traced loops. Bottom: High-pass – filtered image of the partial image used for manual tracing. White means enhanced flux. The regions A and B are shown in more detail in Figures 7 and 8.

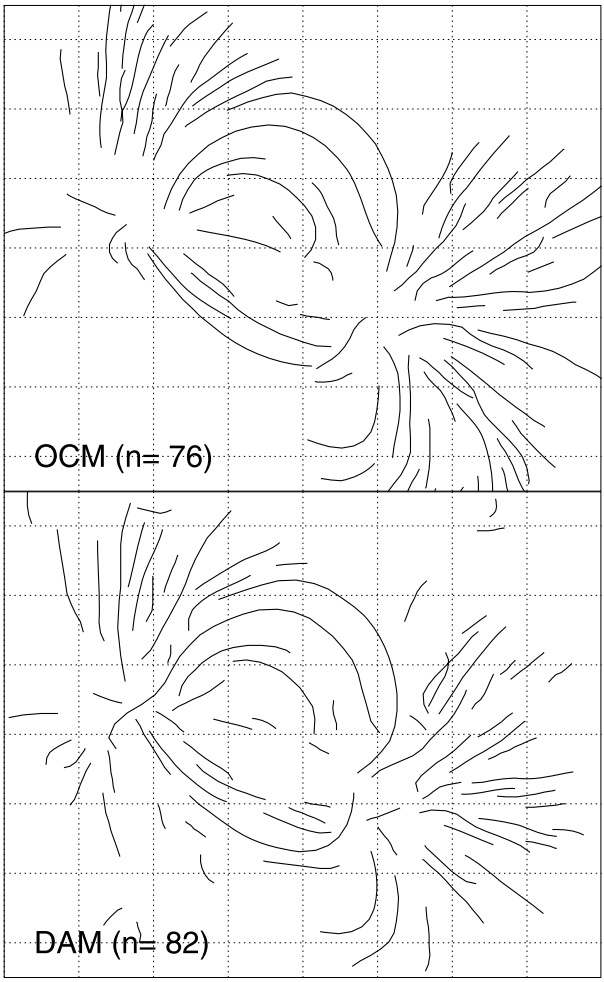

Figure 3 Top: Automated tracing of 76 loops with the OCM code. Bottom: Automated tracing of 82 loops with the DAM code. The coordinate grid has a step of 100 image pixels.

for a suitable number of loops (L), ignoring the shortest loop segments with lengths of $\lesssim 20$ pixels, to suppress unwanted nonloop structures (such as the reticulated *moss* structure seen in the central part of the TRACE image).

3.3.1. Automated Tracing with the OCM Code

The result of the OCM code is shown in Figure 3 (top), which in addition to the TRACE image made use of a near-simultaneous magnetogram from SOHO/MDI. The OCM code finds 76 loops, which appear fairly smooth and are mostly nonintersecting.

3.3.2. Automated Tracing with the DAM Code

The result of the DAM code, shown in Figure 3 (bottom), is very similar to the result of the OCM code (Figure 3, top). The DAM code also finds a similar number of loops ($n = 82$), located at almost the same locations as those of the OCM code.

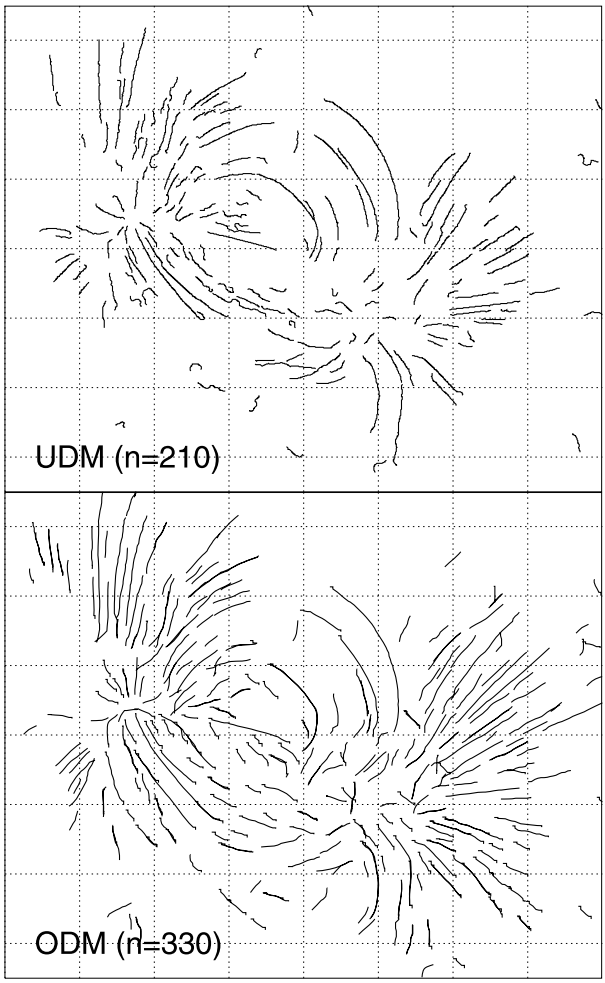

Figure 4 Top: Automated tracing of 210 loops with the UDM code. Bottom: Automated tracing of 330 loops with the ODM code.

3.3.3. Automated Tracing with the UDM Code

The result of the UDM code, shown in Figure 4 (top), comprises $n = 210$ loop segments. The algorithm (written in C) was run with a σ of 1.0 (where σ pertains to the Gaussian kernel used in convolving the test image) and a minimum loop length of $L_{\rm loop} > 20$ pixels. (We add also a note of caution to future users of this code that the C code has a column – row indexing of the image matrix that is opposite to that of IDL, and thus it produces an output with inverted x and y coordinates.)

3.3.4. Automated Tracing with the ODM Code

A result of the ODM code, shown in Figure 4 (bottom), gives a total of $n = 330$ loop segments. The preprocessing was done with high-pass filtering with boxcars of sizes three, five, and seven pixels. The run of the ODM code had the following parameter settings: $N_{\rm macro} = 10$ pixels for the box size of the search of starting points, $N_{\rm box} = 10$ for the size of

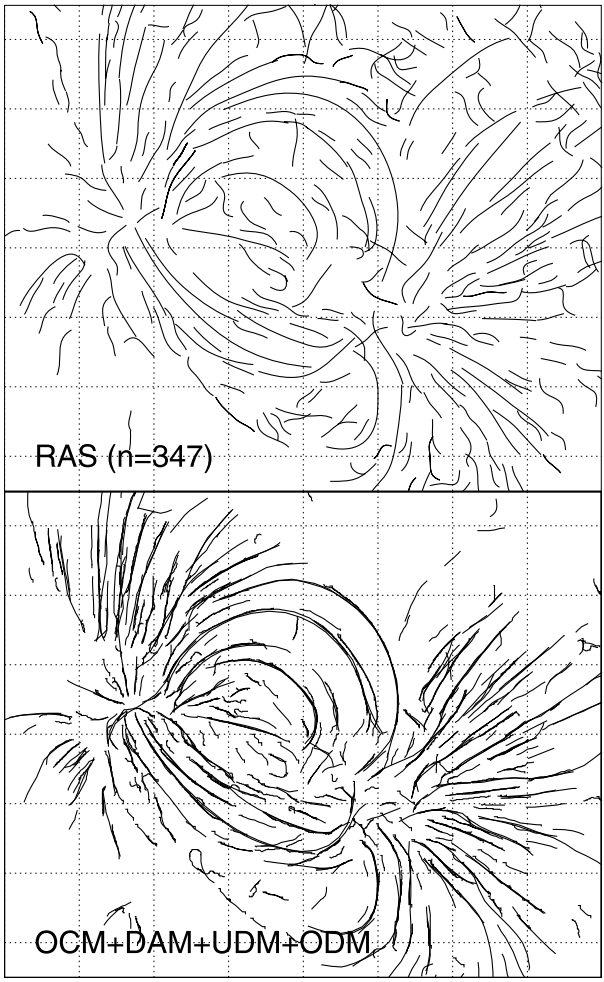

Figure 5 Top: Automated tracing of 347 loops with the RAS code. Bottom: Overlay of OCM + DAM + UDM + ODM codes.

the moving box, $N_{\rm step} = 5$ pixels for the stepping along loops, $N_{\rm dist} = 2$ pixels for the minimum distance to an adjacent loop, $\Delta_{\rm max} = 1.0$ pixel for the maximum deviation of peaks considered in the (directional) linear regression fit, $L_{\rm loop} = 30$ pixels for the minimum length of loop segments, and $F_{\rm thresh} = 0.3$ DN for the minimum flux threshold. The output of the ODM code is similar to the UDM code, showing many more short loop segments than the OCM and DAM codes (shown in Figure 3).

3.3.5. Automated Tracing with the RAS Code

The result of the RAS code, shown in Figure 5 (top), gives a total of $n = 347$ loop segments. The parameters used for this run were $h_{\rm max} = 5$ pixels for the maximum distance between ridgels, $r_{\rm min} = 30$ pixels for the minimum curvature radius, and $a_{\rm max} = 25°$ for the maximum deviation of fit normal to the ridgel orientation. The output of this code reveals the largest number of details, identifying a similar number of loop structures in the central region but more segments in the outer (northwestern) part of the active region than the other four codes combined together (Figure 5, bottom).

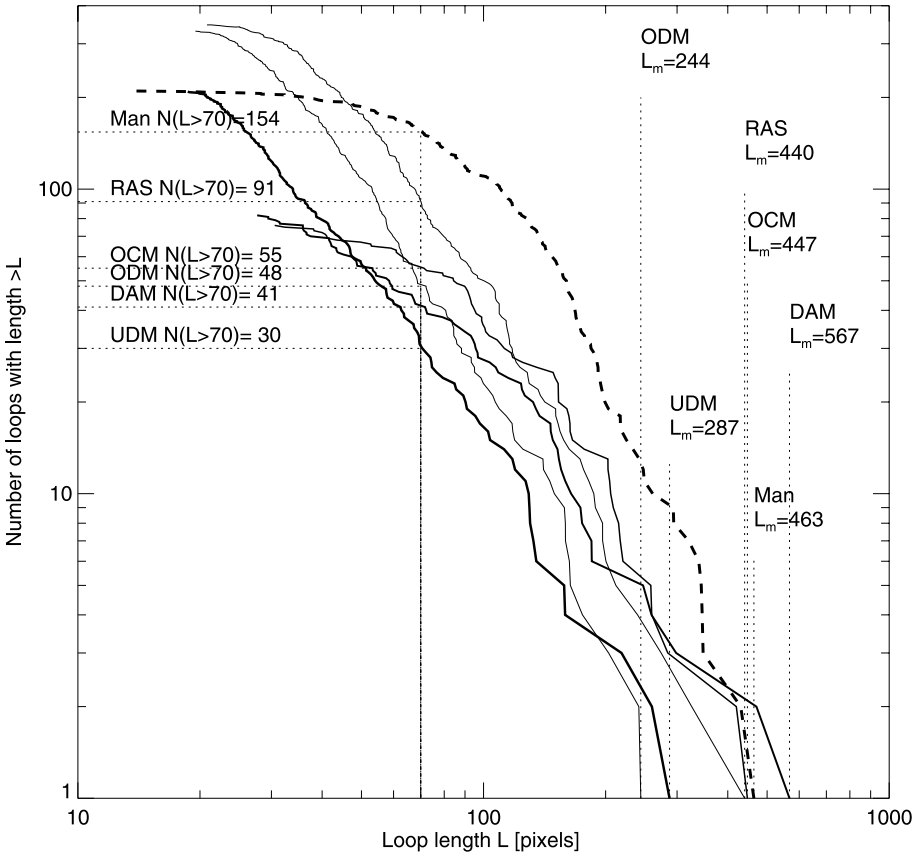

Figure 6 The cumulative size distribution $N(> L_{\rm loop})$ of detected loop lengths L shown for all five codes, plus the manual method. The maximum detected lengths are indicated on the right side, from $L_{\rm m} = 244$ pixels for the ODM code to $L_{\rm m} = 567$ pixels for the DAM code. The relative number of detected loop segments with a length longer than $L = 70$ pixels is shown on the left, ranging from $N(L > 70) = 30$ for the UDM code to $N(L > 70) = 91$ for the RAS code.

3.4. Quantitative Comparison of Automated Tracing Codes

We compare now some quantitative criteria among the different codes, such as the cumulative distribution of loop lengths, the maximum and median detected loop lengths, and the completeness of detection.

3.4.1. Cumulative Distribution of Loop Lengths

A first characterization is the length L of the identified loops. In Figure 6 we show the cumulative distribution $N(L > L_{\rm loop})$ of the number of loops that are longer than a given loop length $L_{\rm loop}$. This cumulative distribution is simply constructed by counting all loops within the given length ranges.

The obtained cumulative distributions show different functional forms; some are closer to a power-law distribution (*e.g.*, the UDM code), whereas others are almost exponential

Table 1 Parameters of cumulative loop length distributions (Figure 6).

Numerical code	Power-law slope β	Ratio of maximum detected loop length $L_m/L_{m,0}$	Ratio of median detected loop length $L_{med}/L_{med,0}$	Ratio of number of detected loops $N(>70)/N_0(>70)$
Manual	-2.8 ± 0.04	1.00	1.00	1.00
OCM	-2.3 ± 0.1	0.97	0.93	0.35
DAM	-2.0 ± 0.1	1.22	0.67	0.27
UDM	-2.5 ± 0.1	0.62	0.31	0.19
ODM	-3.2 ± 0.2	0.53	0.37	0.36
RAS	-2.8 ± 0.04	0.95	0.47	0.59

(*e.g.*, the ODM code). The difference in the cumulative distributions is mostly at short loop lengths, say at $L \lesssim 100$ pixels, whereas they are nearly power-law-like at larger lengths of $L \gtrsim 100$ pixels. The manually traced loops exhibit an almost exponential distribution (dashed curve in Figure 6). It would be premature to interpret the functional shape of these distributions, because they contain a large number of incomplete loop segments and thus may be quite different from the corresponding distributions of complete loop lengths. Most distributions have a similar power-law slope of $\beta \approx -2.0, \ldots, -3.2$ (Table 1) in the loop length range of $L > 100$ pixels, which corresponds to slopes of $\alpha = \beta - 1 \approx -3.0, \ldots, -4.2$ for the differential frequency distribution $N(L_{loop}) \propto L^{\alpha}$ of loop sizes. The automatically traced loops have a distribution with a similar slope as the manually traced selection of loops ($\beta = -2.8, \alpha = -3.8$). However, our loop-length distributions obtained here are somewhat steeper than those inferred in earlier works [*e.g.*, $\alpha \approx -2.1$ for nanoflare loops with lengths in the range of $L \approx 2-20$ Mm (Aschwanden *et al.*, 2000b) and $\alpha \approx 2.9 \pm 0.2$ for TRACE 171 Å loops with lengths of $L \approx 5-50$ Mm, $\alpha \approx 2.6 \pm 0.2$ for TRACE 195 Å loops with lengths of $L \approx 6-70$ Mm, and $\alpha \approx 2.3 \pm 0.3$ for *Yohkoh*/SXT loops with lengths of $L \approx 2-10$ Mm (Figure 5 in Aschwanden and Parnell, 2002)], probably because of the detection of partial, and thus incomplete, loop segments. Regarding the absolute number of loops detected in an image, we note that it depends on a number of variables in the pre- and postprocessing stage, as well as on the particular algorithms of connecting loop segments.

3.4.2. Maximum Detected Loop Lengths

The maximum detected loop lengths (indicated on the right side of Figure 6) are an indication of the robustness of the automated tracing codes to trace long loops, despite the unavoidable interruptions or intersections caused by crossing background loops, or resulting from the weaker fluxes near loop tops, since the electron density in the upper corona (in particular above one hydrostatic density scale height) fades out exponentially with altitude.

The longest loop was traced with the DAM code with a length of $L_m = 567$ pixels (≈ 205 Mm), followed by OCM with $L_m = 447$ pixels (≈ 162 Mm) and RAS with $L_m = 440$ pixels (≈ 159 Mm), whereas the UDM code produces only $L_m = 287$ pixels (≈ 104 Mm) and the ODM yields $L_m = 244$ pixels (≈ 88 Mm). If we take the longest manually traced loop as a normalization, with $L_0 = 463$ Mm (≈ 167 Mm), the various codes achieved the following relative ratios: $L_m/L_{m,0} = 1.22$ for DAM, 0.97 for OCM, 0.95 for RAS, 0.62 for UDM, and 0.53 for ODM. These ratios are a good measure of the code's

ability to trace the longest loops. Figure 3 (bottom panel) actually reveals that the longest loop traced with DAM probably consists of two loops that have been erroneously connected (*i.e.*, the falsely connected loops had end points that were close to each other and had similar slopes). The second-longest loop traced with DAM (see the distribution in Figure 6) actually matches closely the longest traced manual loop. Thus, three of the codes (OCM, DAM, and RAS) are capable of tracing loops as long as the manually traced ones, and two codes fall a bit short (UDM and ODM). If we determine the median length of detected loop segments we obtain the following ratios with regard to the manually detected loops (Table 1): $L_{\rm med}/L_{\rm med,0} = 0.93$ (OCM), 0.67 (DAM), 0.60 (ODM), 0.47 (RAS), and 0.31 (UDM).

3.4.3. Completeness of Loop Detection

The number of detected loops obtained here cannot directly be compared to evaluate the completeness of the various codes, because different criteria have been used for the minimum length. To evaluate the relative completeness of the various codes, we compare the number of loops above an identical minimum length, say at an intermediate size of $L = 70$ pixels (≈ 25 Mm). Comparing the cumulative distributions at this intermediate value (see the vertical dashed line at $L = 70$ pixels in Figure 6), we find that the number of manually traced loops is $N_0(L > 70) = 154$, whereas the various codes detected 91 (RAS), 55 (OCM), 48 (ODM), 41 (DAM), and 30 (UDM) loops, which varies from 59% down to 19%. These relative ratios $N(L > 70)/N_0(> 70)$ (Table 1) provide an approximate measure of the relative completeness of loop detection for the various codes. The completeness of the various codes can probably be adjusted by lowering the flux threshold and the minimum length of detected loop segments, as long as the code does not start to pick up random structures in the noise. The fact that all codes pick up no significant structures in the lower left corner of the image indicates that all codes have clearly been adjusted to a safe value above the noise threshold.

3.4.4. Accuracy and Sensitivity of Loop Detection

To investigate the accuracy of automated loop detection, we compare two selected regions in more detail. The two regions are marked in Figure 2, where region A (Figure 7) comprises the x-range 300 – 500 (pixels) and the y-range 600 – 800 of the original image, and region B (Figure 8) comprises the x-range 725 – 925 and the y-range 350 – 550. We visualize the fine structure of the loops with a gray-scale image that includes a superposition of three high-pass filters (using smoothing with a boxcar of three, five, and seven pixels). The high-pass-filtered image regions are shown in the bottom right panels of Figures 7 and 8, and a contour map of the high-pass-filtered image is also overlaid in each of the other panels. The automated curve tracings of the five different codes are displayed with thick black curves in Figures 7 and 8. From the contour maps one can clearly see the noisy ridges of the loops and compare in detail which ridges have been successfully traced, and which ones have been missed or erroneously connected. It appears that each code can be improved to some extent. The OCM, DAM, and UDM codes seem to trace a smaller number of structures than the other codes, and thus the sensitivity could be lowered. The OCM and DAM codes also have a tendency to misconnect the ridges of closely spaced near-parallel loops. The UDM code seems to follow the ridges fairly exactly but has tends to stop at shorter segments of the loops than the other codes (see also the median loop length of $L_{\rm med}/L_{\rm med,0} \approx 0.31$ in Table 1). The ODM code seems to be more complete in tracing all ridges, but it appears to pick up a few spurious loop-unrelated, noisy structures. The RAS code seems to score somewhere between

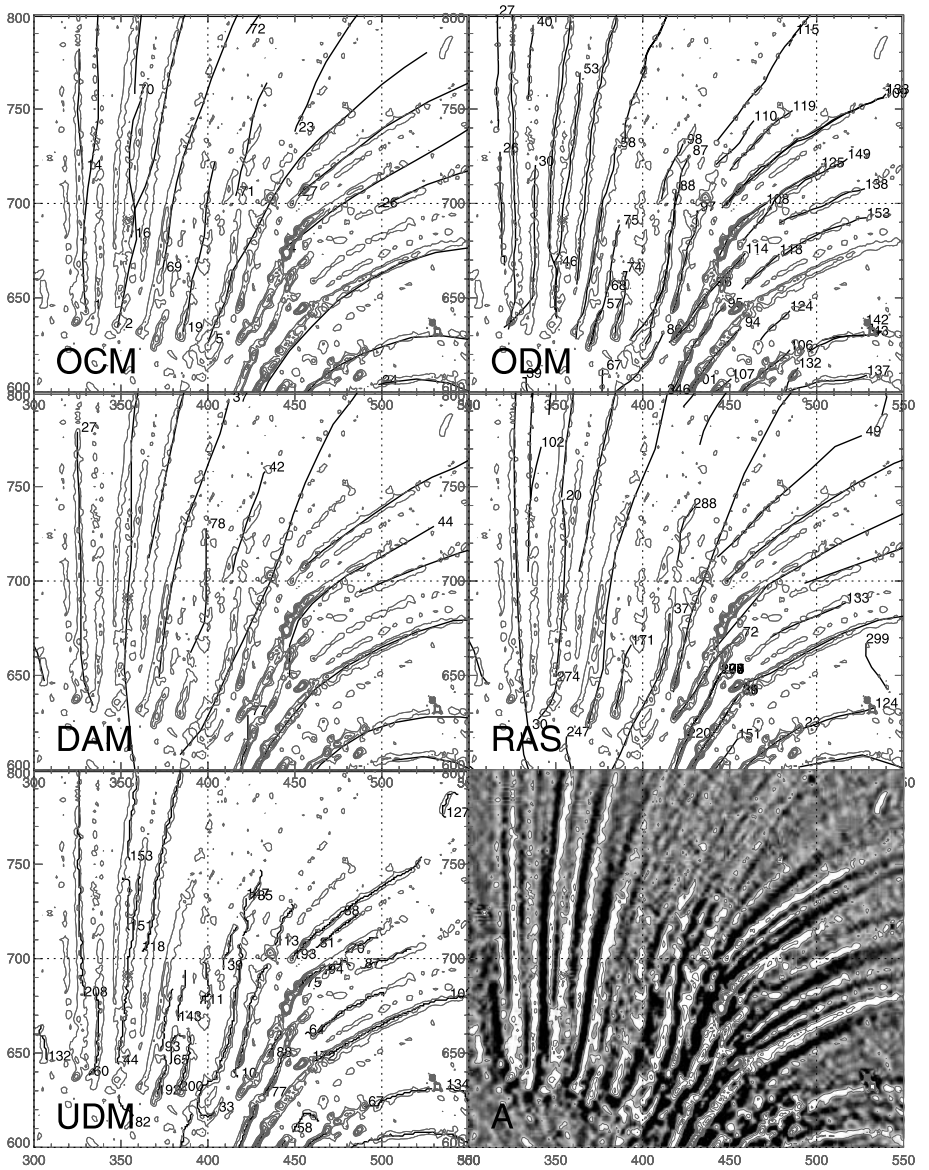

Figure 7 Detailed tracing in image section A (marked in Figure 2: x-range 300 – 550, y-range 600 – 800) by the five different codes. Each panel shows a contour plot of the high-pass-filtered image with contours at 0.5, 1.5, …, 4.5 DN; a gray-scale image is shown in the bottom right panel.

the UDM and ODM codes regarding completeness, but it misconnects a few spurious loop-unrelated, strongly curved structures. We hope that such detailed comparisons, as shown in Figures 7 and 8, stimulate further improvements of the various codes. Some improvements probably can already be achieved with the existing codes by adjusting the built-in control parameters.

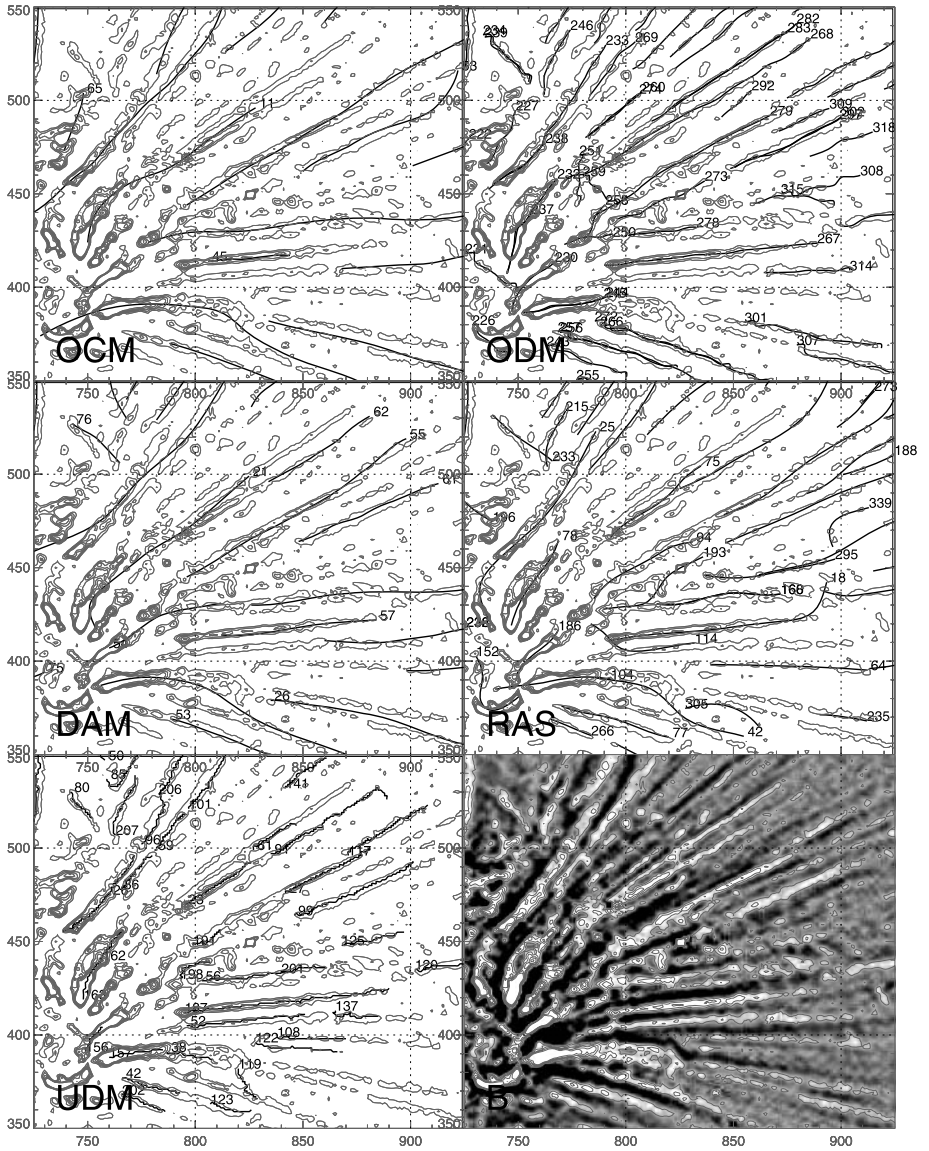

Figure 8 Detailed tracing in image section B (marked in Figure 2: x-range 725 – 925, y-range 350 – 550) by the five different codes and manual tracing. Each panel shows a contour plot of the high-pass-filtered image with contours at 0.5, 1.5, . . . , 4.5 DN.

3.4.5. Computation Speed of Automated Tracing Codes

The computation times of the described codes for processing a single (1000 × 1000) image vary between a few seconds and a few tens of seconds for the described cases. However, it is not meaningful to compare individual performance times at this point, because each code was run on a different machine and most codes are still in a development phase with intermediate test displays, and thus have not yet been optimized for speed. The running

time also scales linearly with the volume of the parameter space that a code is processing. For instance, the ODM code had a running time of 5.9 seconds for the case described in Section 3.3.4 for the given macropixel size (for searching of loop starting points), but it took four times longer if the macropixel size was halved (compared with a running time of 0.93 seconds for the standard TIME_TEST2 in IDL, with a Mac Power PC 970FX v3.1 processor). We expect that the optimized codes will reach processing times of a few seconds per image.

4. Discussion and Conclusions

Every quantitative analysis of the geometry of a coronal loop requires an accurate measurement of its 2D coordinates in solar images. If we could localize an isolated loop in a solar image, the problem would be trivial, because the cross section along the loop axis could be fitted with some generic function (*e.g.*, a Gaussian), and this way the loop centroid positions (x_i, y_j) could accurately be measured (with subpixel accuracy) for a number of spline points $i = 1, \ldots, N$ along the loop length, with the two endpoints $i = 0$ and $i = N$. Such accurate loop coordinates (x_i, y_i), which merely mark the projected position of the loop, could then be used to reconstruct the 3D geometry, *e.g.*, by triangulating the same loop with two stereoscopic spacecraft, which yield two different projections, $(x_i^{\rm A}, y_i^{\rm A})$ and $(x_i^{\rm B}, y_i^{\rm B})$, from which the 3D geometry (x_i, y_i, z_i) can be computed by simple trigonometry. Of course, this assumes that projection effects resulting from transparent loop structures are insignificant (*i.e.*, that each loop in the image projection represents a physical coronal loop).

In reality, however, there is no such thing as an "isolated loop," but each loop is observed on top of a background that consists of some $\approx 10^3$ other coronal loops along each line of sight. [See the statistical model of Composite and Elementary Loop strands in a Thermally Inhomogeneous Corona (CELTIC) that specifies loop structures and their coronal background in a self-consistent way (Aschwanden, Nightingale, and Boerner, 2007).] This means that the identification of a single loop becomes very ambiguous, and a visual definition may not be sufficient. Mostly for this reason, there is a demand for an automated loop-tracing code that is based on mathematical criteria rather than visual judgment.

In this paper, we explored for the first time systematically the performance of five such automated loop-tracing codes, developed by four independent groups. Because all five codes work fully automatically without human interaction, they should be able to recover the existing information on loop coordinates in the most objective way and, ideally, should converge to the same result within the uncertainty of the data noise. The comparison made here revealed significant differences in the performance of these five codes; for instance, the maximum loop length was detected between 53% and 122% of that obtained from manual/visual tracing, or the median length varied between 31% and 93% of that obtained from manual tracing. Also the detection efficiency or completeness varies substantially (*i.e.*, the number of detected loops with intermediate to large sizes varied between 19% and 93% of the manually traced reference set). Of course, the manual/visual tracing should not be the ultimate arbiter in the evaluation of automated numerical codes, but it provides at least an educated guess of how many structures are to be expected, based on the quantitative output from visual pattern recognition. One could construct an artificial test image with a well-defined number and distribution of loops, but such artificial test data are only useful if they accurately mimic the real data regarding morphological structures, the distributions of parameters, and data noise.

These experiments, therefore, can be used to adjust the control parameters of flux sensitivity, minimum length, minimum separation, minimum curvature, *etc.*, of each code, so

that they can be run more consistently versus each other. Once all codes are tuned to the same sensitivity, they should produce about the same number of detected structures, or at least a consistent cumulative distribution of loop lengths. The control parameters can then iteratively be adjusted until all codes produce the same results within the uncertainties of the data noise, unless some codes have an inherent incapability to achieve the same task, or use external (*a priori*) information from physical models.

In the end, even when all codes converge to the same optimum solution, we might learn what the true limitations of automated loop recognition are. We know already that the top parts of large coronal loops are untracable, in particular when only one single temperature filter is used, because the emission measure drops below the noise threshold as a consequence of gravitational stratification (or other physical mechanisms). Also the footpoints of the loops may not be visible because of the temperature drop from the corona toward the transition region, but the cooler part of the loop footpoints may be only a negligible small fraction of the entire loop length. Another limitation is the complexity of the background, which can disrupt loop tracing at countless locations beyond repair. The SDO/AIA data will greatly improve the detection on nonisothermal loops owing to its wider temperature coverage. Also, the use of time sequences of images will render loop detection more redundant, and thus more robust. Nevertheless, even if we are only able to produce reliable measurement of partial loop segments, we will have stringent constraints for testing of theoretical (*e.g.*, nonlinear force-free) magnetic field models (*e.g.*, Schrijver *et al.*, 2006), for correcting magnetic field solutions (*e.g.*, Gary and Alexander, 1995), and for stereoscopic reconstruction of the 3D geometry of loops and magnetic field lines (*e.g.*, Wiegelmann and Inhester, 2006).

Acknowledgements We acknowledge helpful discussions with David Alexander, Karel J. Schrijver, Marc DeRosa, Paulett Liewer, and Eric DeJong. The work benefitted from the stimulating discussions at the 3rd Solar Image Processing Workshop in Dublin, Ireland, 6 – 8 September 2006 and at the 5th SECCHI Consortium Meeting in Orsay, Paris, 5 – 8 March 2007. Part of the work was supported by NASA TRACE Contract No. NAS5-38099 and NASA STEREO/SECCHI Contract No. N00173-02-C-2035 (administrated by NRL).

References

Aschwanden, M.J.: 2004, *Physics of the Solar Corona – An Introduction*, Praxis, Chichester, and Springer, Berlin.
Aschwanden, M.J.: 2005, *Solar Phys.* **228**, 339.
Aschwanden, M.J., Alexander, D.: 2001, *Solar Phys.* **204**, 93.
Aschwanden, M.J., Nightingale, R.W.: 2005, *Astrophys. J.* **633**, 499.
Aschwanden, M.J., Parnell, C.E.: 2002, *Astrophys. J.* **572**, 1048.
Aschwanden, M.J., Nightingale, R.W., Alexander, D.: 2000, *Astrophys. J.* **541**, 1059.
Aschwanden, M.J., Nightingale, R.W., Boerner, P.: 2007, *Astrophys. J.* **656**, 577.
Aschwanden, M.J., Newmark, J.S., Delaboudinière, J.P., Neupert, W.M., Klimchuk, J.A., Gary, G.A., Portier-Fornazzi, F., Zucker, A.: 1999a, *Astrophys. J.* **515**, 842.
Aschwanden, M.J., Fletcher, L., Schrijver, C., Alexander, D.: 1999b, *Astrophys. J.* **520**, 880.
Aschwanden, M.J., Alexander, D., Hurlburt, N., Newmark, J.S., Neupert, W.M., Klimchuk, J.A., Gary, G.A.: 2000a, *Astrophys. J.* **531**, 1129.
Aschwanden, M.J., Tarbell, T., Nightingale, R., Schrijver, C.J., Title, A., Kankelborg, C.C., Martens, P.C.H., Warren, H.P.: 2000b, *Astrophys. J.* **535**, 1047.
Aschwanden, M.J., DePontieu, B., Schrijver, C.J., Title, A.: 2002, *Solar Phys.* **206**, 99.
Carcedo, L., Brown, D., Hood, A., Neukirch, T., Wiegelmann, T.: 2003, *Solar Phys.* **218**, 29.
Gary, G.A., Alexander, D.: 1995, *Solar Phys.* **186**, 123.
Inhester, B., Feng, L., Wiegelmann, T.: 2007, *Solar Phys.* doi:10.1007/S11207-007-9027-1.
Lee, J.K., Newman, T.S., Gary, G.A.: 2004, In: *Proc. 17th Int. Conf. on Pattern Recognition (ICPR)*, Cambridge, United Kingdom, 315. doi:10.1007/S11207-007-9027-1.

Lee, J.K., Newman, T.S., Gary, G.A.: 2006a, *Pattern Recognit.* **39**, 246.
Lee, J.K., Newman, T.S., Gary, G.A.: 2006b, In: *Proc. 7th IEEE SSIAI*, Denver, CO, 91
Lindeberg, T.: 1998, *Int. J. Comput. Vision* **30**, 117.
Parent, P., Zucker, S.: 1989, *IEEE Trans. Pattern Anal. Mach. Intel.* **11**, 823.
Raghupathy, K.: 2004, Ph.D. thesis, Cornell University.
Schrijver, C.J., DeRosa, M., Metcalf, T.R., Liu, Y., McTiernan, J., Regnier, S., Valori, G., Wheatland, M.S., Wiegelmann, T.: 2006, *Solar Phys.* **235**, 161.
Smith, M.: 2005, http://www.mssl.ucl.ac.uk/twiki/bin/view/SDO/LoopRecognition.
Steger, L.H.: 1996, Technical Report FGBV-96-03, Forschungsgruppe Bildverstehen (FG BV), Informatik IX, Technische Universität München, Germany.
Strous, L.H.: 2000, http://www.lmsal.com/~aschwand/stereo/2000_easton/cdaw.html.
Wiegelmann, T., Inhester, B.: 2006, *Solar Phys.* **236**, 25.
Yan, Y., Aschwanden, M.J., Wang, S.J., Deng, Y.Y.: 2001, *Solar Phys.* **204**, 29.